

UNIVERZA V LJUBLJANI
FAKULTETA ZA RAČUNALNIŠTVO IN INFORMATIKO

Rok Banko

**Platforma za borzo kmetijske
mehanizacije**

DIPLOMSKO DELO
NA UNIVERZITETNEM ŠTUDIJU

MENTOR: doc. dr. Rok Rupnik

Ljubljana, 2015

To delo je ponujeno pod licenco Creative Commons Priznanje avtorstva-Deljenje pod enakimi pogoji 2.5 Slovenija (ali novejšo različico). To pomeni, da se tako besedilo, slike, grafi in druge sestavine dela kot tudi rezultati diplomskega dela lahko prosto distribuirajo, reproducirajo, uporabljajo, priobčujejo javnosti in predelujejo, pod pogojem, da se jasno in vidno navede avtorja in naslov tega dela in da se v primeru spremembe, preoblikovanja ali uporabe tega dela v svojem delu, lahko distribuira predelava le pod licenco, ki je enaka tej. Podrobnosti licence so dostopne na spletni strani creativecommons.org/licenses/by-sa/2.5/si/ ali na Inštitutu za intelektualno lastnino, Streliška 1, 1000 Ljubljana.



Izvorna koda diplomskega dela, njeni rezultati in v ta namen razvita programska oprema je ponujena pod licenco GNU General Public License, različica 3 (ali novejša). To pomeni, da se lahko prosto distribuira in/ali predeluje pod njenimi pogoji. Podrobnosti licence so dostopne na spletni strani <http://www.gnu.org/licenses/>.

Besedilo je oblikovano z urejevalnikom besedil \LaTeX .

Fakulteta za računalništvo in informatiko izdaja naslednjo nalogo:

Tematika naloge:

V diplomski nalogi predstavite potek razvoja platforme za borzo kmetijske mehanizacije od ideje do rešitve. Poleg spletnega vmesnika, omogočite uporabnikom iskanje tudi preko mobilnih naprav. Osredotočite se na uporabo lokacijskih podatkov za hitrejše iskanje oglasov ter uporabnikom ponudite sistem pametnega ujemanja z njihovimi oglasi. V zaključnem delu predstavite prednosti platforme, v primerjavi s podobnimi obstoječimi rešitvami, in podajte možne poti nadaljnega razvoja.

IZJAVA O AVTORSTVU DIPLOMSKEGA DELA

Spodaj podpisani Rok Banko, z vpisno številko **63060016**, sem avtor diplomskega dela z naslovom:

Platforma za borzo kmetijske mehanizacije

S svojim podpisom zagotavljam, da:

- sem diplomsko delo izdelal samostojno pod mentorstvom doc. dr. Roka Rupnika,
- so elektronska oblika diplomskega dela, naslov (slov., angl.), povzetek (slov., angl.) ter ključne besede (slov., angl.) identični s tiskano obliko diplomskega dela
- soglašam z javno objavo elektronske oblike diplomskega dela na svetovnem spletu preko univerzitetnega spletnega arhiva.

V Ljubljani, dne 15. junija 2015

Podpis avtorja:

Zahvaljujem se vsem, ki so me tekom študija podpirali in spodbujali, posebno pa staršem, bratu in Hani za več kot potrebno motivacijo. Zahvala gre tudi doc. dr. Roku Rupniku za pomoč pri izdelavi diplomske naloge.

Kazalo

Seznam uporabljenih kratic

Povzetek

Abstract

1	Uvod	1
2	Razvojna orodja in tehnologije	3
2.1	Razvojna orodja	3
2.2	Ogrodje Zend Framework 2 in knjižnice	6
2.3	Tehnologije	7
3	Analiza in načrtovanje platforme	9
3.1	Diagram primerov uporabe	9
3.2	Načrtovanje podatkovne baze	11
4	Implementacija	15
4.1	Arhitektura	15
4.2	Struktura datotečnega sistema	16
4.3	Mobilna aplikacija	19
5	Opis uporabniškega vmesnika	23
5.1	Administratorski del	23
5.2	Uporabniški del	23

KAZALO

6	Uporaba lokacije in pametno ujemanje oglasov	29
6.1	Izkoriščanje lokacije	29
6.2	Pametno ujemanje oglasov	31
7	Postavitev strežnika in zagon aplikacije	35
7.1	Postavitev strežnika	35
7.2	Subversion in objava aplikacije	35
8	Sklepne ugotovitve	37
8.1	Možnosti nadaljnjega razvoja	38

Seznam uporabljenih kratic

kratica	angleško	slovensko
HTML	Hyper Text Markup	jezik za označevanje nadbesedila
CSS	Cascading Style Sheets	kaskadne stilske podloge
AJAX	Asynchroneus JavaScript and XML	asinhroni JavaScript in XML
SQL	Script Query Language	strukturirani povpraševalni jezik za podatkovne baze
HTTP	Hyper Text Transfer Protocol	protokol za prenos nadbesedil
JSON	JavaScript Object Notation	Notacija javascript objektov
ZF2	Zend Framework version 2	Zend Framework različica 2
URL	Uniform resource locator	enolični krajevnik vira
GPS	Global positioning system	Globalni sistem pozicioniranja
MVC	Model-view-controller	Model-pogled-krmilnik
C2C	Customer to customer	potrošnik - potrošnik
SVN	Subversion code repository	repozitorij kode Subversion
WAMP	Windows, Apache, MySQL, PHP server	Windows, Apache, MySQL, PHP strežnik

Povzetek

V diplomskem delu prikažemo razvoj in postavitev platforme za borzo kmetijske mehanizacije z aktualnimi orodji. Glavni prednosti platforme sta uporaba odjemalčeve lokacije, kar ponuja lažje lociranje bližnjih oglasov, in pametnega ujemanja oglasov z oddanimi oglasi uporabnika.

V prvem delu prikažemo načrtovanje podatkovne baze, glede na predvidene primere uporabe, v orodju MySQL Workbench in njeno postavitev z uporabo baze MySQL. Nato podamo opis razvoja platforme z uporabo MVC koncepta v ogrodju Zend Framework 2. Tu se obračamo na tehnologije kot so HTML, PHP, AJAX, JSON in Javascript. Sledi še hiter prelet uporabe različnih knjižnic, ki so naš razvoj poenostavile in pospešile. V nekaj odstavkih predstavimo in opišemo tudi razvoj mobilne aplikacije z uporabo PhoneGap-a.

Drugi del temelji na uporabi uporabnikove lokacije in njeno vključitev v samo platformo. Podamo opis algoritma za pametno ujemanje oglasov in prikaz rezultatov na uporabniškem profilu.

Delo zaključimo z idejami in možnostmi nadaljnjega razvoja platforme ter hitro primerjavo s podobnimi, že obstoječimi, rešitvami.

Abstract

In the thesis we present the development of a platform for exchanging agricultural machinery. In the process of planning and implementing the project we take current tools and follow a modern concept of development. The two main advantages of the platform are the use of customer location for easier and faster searching of items and a process called "smart matches" that is periodically checking if the user active items have any opposing type items that could be interesting for the user.

The first part is comprised of database planning, based on the use case diagram, in MySQL Workbench and its realization in MySQL database. We also describe the platform implementation in Zend Framework 2 using the Model-View-Controller concept. Technologies like HTML, PHP, AJAX, JSON and JavaScript are widely used here. The first part ends with a brief description of libraries used to help us shorten the implementation time and a few words about the programming of a mobile application with PhoneGap tool.

The second part is focused on the two main advantages of the platform. Client position utilization in our platform and items smart matches algorithm are presented within this part.

The work concludes with ideas and possibilities for further development of the platform and a quick comparison with similar, existing solutions.

Poglavje 1

Uvod

Dostop do spleta in storitev, ki jih le ta ponuja, je danes možen že praktično povsod. Zaradi enostavnosti dostopa in uporabnosti se je tako razširil preko vseh človekovih dejavnosti tudi do panog, kjer se tehnologija uveljavlja počasneje. Kmetijstvo oz. agrikultura je področje, kjer se, z menjavanjem generacij, uporaba svetovnega spletna vedno hitreje uveljavlja. Kmetovalec tako ni več odvisen le od pomoči sosedov in najbližje trgovine s kmetijsko mehanizacijo, ampak se je njegov trg razširil globalno.

Preko oglasnikov je mogoče kupovati in prodajati kmetijsko mehanizacijo, ne obstaja pa platforma, ki bi kmetovalcem omogočala tudi najem ali oddajo njihovih storitev drugim kmetovalcem. Cene traktorjev, balirk, cepilnikov in druge mehanizacije so visoke, zato bi bil njihov najem ali oddajanje smotrna poteza. Stroji bi se zaradi večje uporabe prej odplačali, kmetje pa bi lahko z najemom storitev hitreje opravili svoje delo.

Analiza storitev, ki so trenutno na voljo, je pokazala, da so na spletu večinoma prisotne le borze tipa prodaj in kupi. Pri storitvah, ki nudijo najem, gre v veliki meri za podjetja, ki svoje stroje posojajo za določen čas. V času analize ni bilo mogoče najti borze, ki bi omogočala C2C izposojanje in nudenje storitev ali strojev. Sploh pa ne take, ki bi bila specializirana za kmetijstvo in kmetijsko mehanizacijo. Najbližjo podobnost z iskanim predstavlja portal bolha.com, ki pa nudi najemanje in oddajanje le za turistične objekte.

Kot rešitev smo razvili platformo, ki je specializirana za kmetijstvo in ponuja možnost oddajanja ter najemanja strojev. Platformo smo obogatili

z lokacijskimi storitvami, kar omogoča hitrejšo iskanje oglasov glede na oddaljenost od uporabnika. Poleg tega se uporabnikom, ki sami oddajo oglas, priporoči oglas nasprotnega tipa na podlagi pomembnosti in oddaljenosti. Uporaba platforme je možna tudi preko mobilnih naprav, saj se spletna stran prilagodi velikosti zaslona.

V prvem delu predstavimo programska orodja, ogrodje Zend Framework 2 in tehnologije za razvoj platform. Nato se fokusiramo na razvoj in samo delovanje aplikacije. Za konec pa podamo še podrobnejši opis pametnega ujemanja oglasov in sklepne ugotovitve.

Poglavje 2

Razvojna orodja in tehnologije

V tem poglavju bomo predstavili orodja, s katerimi smo platformo razvijali, in sicer PhpStorm, MySQL Workbench, Notepad++, WAMP, PhoneGap in Firebug. Za verzioniranje kode smo uporabili orodje Subversion z Windows klientom TortoiseSVN. Za sprotno pregledovanje delovanja in ugotavljanja napak izdelka smo vzeli najbolj pogosto uporabljene moderne brskalnike - Firefox, Chrome in Internet Explorer.

Za lokalno razvijanje smo uporabili WAMP strežnik. Uradno stran pa smo postavili s pomočjo Apache Web Server-ja, PHP in podatkovne baze MySQL.

2.1 Razvojna orodja

2.1.1 Notepad++

Notepad++^[1] je napredni tekstovni urejevalnik, ki je pisan na kožo vsem programerjem, saj ponuja veliko število funkcij. Njegova majhnost in hiter zagon sta idealna za majhne popravke kode. Glede na končnico datoteke samodejno prepozna vrsto programskega jezika in obarva sintakso. Omogoča tudi primerjanje vsebine datotek, iskanje po vseh odprtih datotekah, več vrstično iskanje in urejanje, prikaz kontrolnih znakov in mnogo drugega. Zmogljivost lahko razširimo tudi s številnimi vtičniki.

Resnega razvoja v njem sicer ni pričakovati, saj smo med samim pisanjem kode osredotočeni le na odprto datoteko. Večja (in dražja) orodja nam med drugim ponujajo samodejno dopolnjevanje, prej pripravljene vzorce kode in analizo kode pred potencialnimi programskimi hrošči.

Orodje je prosto dostopno na spletu in je na voljo pod licenco GPL.

2.1.2 PhpStorm

PhpStorm[2] je izdelek podjetja JetBrains. Predstavljen je kot najbolj pаметen urejevalnik PHP kode. Zaradi vgrajenega dopolnjevanja kode, hitre navigacije po kodi, hitrega in varnega refactoringa¹, lažjega odpravljanja hroščev ter podpore raznim ogrođjem je nadvse primeren za resen razvoj. Vgrajeno orodje za delo s sistemi za verzioniranje kode in njeno objavlanje, na uradno stran, nam prihranita veliko časa pri delu. Pozna vse jezike, ki se uporabljajo pri izdelavi spletnih aplikacij, in njihovo sintakso tudi obarva ter analizira že med samim pisanjem. Prepozna objekte v PHP kodi, tako da je delo z njimi zelo enostavno.

Razvoj zaradi teh značilnosti poteka zelo hitro. Program nas opozori na napako oz. nepravilnost takoj, ko zaključimo s pisanje stavka. Na ta način nas večkrat obvaruje pred izgubo dragocenega časa pri iskanju vzroka napak.

Licenca za komercialne uporabnike se giblje okoli 200€, vendar je za študente, učitelje in odprtokodne projekte brezplačna.

2.1.3 MySQL Workbench

MySQL Workbench[3] je skupek orodij za arhitekta, razvijalce in administratorje podatkovnih baz. Za arhitekta so na voljo grafična orodja za gradnjo, podatkovno modeliranje in načrtovanje podatkovnih baz. Orodje nam preko avtomatiziranega postopka naš podatkovni model zgradi v podatkovni bazi MySQL in ga ob kasnejših dodelavah sinhronizira. Razvijalci lahko grafično sestavljajo poizvedbe SQL in analizirajo njihovo izvajanje. Administratorji pa lahko preko grafičnega vmesnika spremljajo delovanje baze in jo upra-

¹refactoring - process preimenovanja, združevanja ali razbijanja kode s ciljem lažjega vzdrževanja in razumevanja. Pri tem ne pride do spreminjanja delovanja.

vljajo. Poleg tega lahko s tem orodjem tudi migrirajo podatke iz drugih baz na MySQL podatkovno bazo.

Orodje je prosto dostopno na spletu.

2.1.4 PhoneGap

PhoneGap[8] je orodje za razvijanje mobilnih aplikacij z uporabo tehnologij HTML, CSS in JavaScript. Sočasno omogoča razvoj na operacijskih sistemih Windows Phone, Andorid in iOS z enotno kodo. Aplikacija uporablja Apache Cordova[9] API vmesnik, ki so ga pri Apache-u razvili kot univerzalni vmesnik med aplikacijo in telefonom.

PhoneGap aplikacijo se namesti na razvojni računalnik in na mobilne telefone/tablice. Razvojni računalnik deluje kot strežnik, ki napravam v istem brezžičnem omrežju pošilja prevedeno aplikacijo. Razvijalec lahko spremembe, ki jih naredi v kodi, takoj zazna na vseh mobilnih napravah. Ko je mobilna aplikacija končana, se izvirno kodo naloži na Adobe PhoneGap Build[10], kjer se v oblaku prevede za tri mobilne sisteme - iOS, Andorid in Windows Phone. Aplikacija je takoj na voljo za objavo v različnih mobilnih trgovinah.

Za odprtokodne projekte in en zaseben projekt do velikosti 50MB je orodje brezplačno.

2.1.5 Subversion

Subversion (krajše SVN)[11] je odprtokodni sistem na nadzor različic izvirne kode. Zasnovan je bil že leta 2000, v zadnjem desetletju pa je doživel neverjeten uspeh in bil široko sprejet v odprtokodnih krogih ter poslovnem svetu. Subversion se razvija pod okriljem fundacije Apache Software Foundation.

Verzioriranje izvirne je zelo pomembna prvina programiranja vsakega večjega sistema, saj preprečuje možnost izgube kode, nadzoruje vsako še tako majhno spremembo, ki se jo da kadarkoli razveljaviti, in omogoča sodelovanje večje ekipe na skupnem projektu. Dobra lastnost verzioriranja je tudi možnost povezave z orodji za avtomatizirano prevajanje in testiranje ob vsaki novi različici kode.

Orodje je na voljo brezplačno.

2.1.6 Firebug

Firebug [17] je odprtokodno orodje, ki je primarno namenjeno brskalniku Firefox, čeprav podpira tudi druge brskalnike. Znotraj brskalnika nam ponudi ogromno razvijalskih funkcij, ki vključujejo: analizo in urejanje HTML kode, upravljanje s CSS stili strani, vizualizacijo velikosti in postavitev elementov na strani, spremljanje mrežne dejavnosti, razhroščevanje JavaScripta, konzolo s prikazom napak na strani, upravljanje s spletnimi piškotki in drugo. Akcije so nemudoma vidne na spletni strani, ki jo ima uporabnik naloženo v brskalniku.

Poleg vsega naštetega omogoča tudi izvajanje poljubne JavaScript kode in je primeren tudi za testiranje varnosti spletnih strani[18].

Orodje je na voljo brezplačno.

2.2 Ogrodje Zend Framework 2 in knjižnice

2.2.1 Zend Framework 2

Zend Framework 2 [4] je odprtokodno ogrodje za razvoj spletnih aplikacij in servisov z uporabo jezika PHP (5.3+). Ogrodje temelji na objektno usmerjenem programiranju in izkorišča najnovejše funkcije v jeziku PHP 5.3. Razvito je bilo iz svojega predhodnika Zend Framework 1, ki se lahko pohvali s 15 milijoni prenosov.

Zend Framework 2 ponuja stabilno in zmogljivo MVC ogrodje, abstraktni nivo podatkovnih baz, HTML5 komponente za gradnjo vmesnika in validacijo vnosnih polj, kar smo izkoristili za enostavno postavitev objektno usmerjene rešitve. Z razširitvami in vtičniki lahko hitro vključimo tudi preverjanje pristnosti uporabnikov in nadzor dostopa do vsebin.

Ogrodje je na voljo brezplačno za prenos s svetovnega spleta.

2.2.2 Knjižnice

- **jQuery**[5] - je hitra, majhna in bogata knjižnica JavaScript-a. V njej so spisane pogosto uporabljene funkcije za spreminjanje elementov, odzivanje na dogodke, animacijo in klice API servisov preko AJAX-a, ki jih podpira večina danes modernih, brskalnikov. jQuery je z vsestranskostjo in razširljivostjo spremenil način uporabe JavaScript-a pri milijonih programerjev.
- **jQueryUI**[6] - je posebno izbrana množica vmesniških interakcij, učinkov, gradnikov in tem, zgrajenih na osnovi knjižnice jQuery. Knjižnica je odlična izbira za visoko interaktivne spletne aplikacije, kot tudi za enostavne vmesnike, kjer potrebujete le gradnik za izbiro datuma.
- **Bootstrap**[7] - S širjenjem trenda pametnih telefonov se je pojavila potreba po prilagajanju spletnih vsebin majhnim zaslonom. Trenutno eno izmed najbolj sprejetih ogrodij za to je Bootstrap, ki je nastalo kot izdelek Twitterjevih razvijalcev. Bootstrap vsebuje ikone za vmesnik, pripravljene elemente za opozorila, naslove, odstranjevanje, posebne elemente za gradnjo menijev, tabel, spustne gumbe, vnosna polja, ... Vsi elementi so moderno oblikovani in odzivni na spreminjanje velikosti zaslona. Tehnologija v ozadju je sestavljena iz HTML-ja, CSS-ja in JavaScript-a.

2.3 Tehnologije

2.3.1 HTML in CSS

HTML [12] je označevalni jezik za izdelavo spletnih strani. Njegove korenine segajo v leto 1990, ko se je v CERNU-u porodila ideja o posebni obliki deljenja dokumentov. Avtor, Tim Berners-Lee, je takrat napisal svoj brskalnik in strežnik, ki sta služila prikazu tako zapisanih dokumentov. Javnosti je bil jezik predstavljen leta 1991, ko so definirali prvih 18 značk. Te značke so definirale naslove (<TITLE>), povezave (<A>), sezname (<LISTING>) in še nekaj drugih osnovnih elementov besedila.

V osnovi gre pri HTML-ju za navaden tekstovni dokument, ki s pomočjo značk opisuje strukturo in postavitev končne strani. Brskalnik je orodje, ki nam tako urejeno besedilo pravilno izriše na zaslon glede na sprejeti standard in nam omogoča uporabo povezav za navigacijo med dokumenti. Glede na zgodovino računalništva, ki je znana po velikih spremembah v krajših časovnih obdobjih, je zanimivo, da se je kar polovica začetnih značk obdržala v standardu.

Zadnja večja sprememba je standard HTML5, ki je bil objavljen oktobra 2014 in je prinesel novosti predvsem na področju večpredstavitvenih vsebin, izrisovanja dvodimenzionalnih grafik in pravil za shranjevanja podatkov na uporabnikovem računalniku.

CSS [13, 14] je enostavni mehanizem za oblikovanje besedila na spletnih straneh s pomočjo barv, pisav, razmikov in drugih elementov oblikovanja.

Po začetku razvoja in uporabe HTML-ja širšim množicam ni ustrezalo privzeto oblikovanje besedila. Zato je Håkon Wium leta 1994 poslal organizaciji W3C svoj predlog pravil imenovanih Cascading HTML Style Sheets. Takratna brskalnika Internet Explorer in Netscape sta standard sprejela še pred uradno potrditvijo. Danes skoraj ne obstaja spletna stran, ki pri svojem videzu ne bi uporabljala CSS pravil.

2.3.2 PHP

PHP [15, 16] je odprtokodni programski jezik za gradnjo dinamičnih spletnih vsebin na strežniški strani. Podoben je običajno strukturiranim programskim jezikom. Shranjen je v skriptah, ki na podlagi vhodnih parametrov kot izhod dinamično generirajo spletno stran. Prednost obdelave na strežniku je možnost poganjanja sistemskih ukazov za kompleksnejše operacije ter uporabo in predelavo strežniških vsebin, še preden so te dostavljene klicatelju.

Korenine segajo v leto 1994, ko je dansko kanadski programer Rasmus Lerdorf želel zamenjati nekaj skript, napisanih v Perlu, ki jih je uporabljal za upravljanje svoje spletne strani. Leta 1995 je javnosti izdal izvirno kodo svojih skript pod imenom "PHP Tools".

Poglavje 3

Analiza in načrtovanje platforme

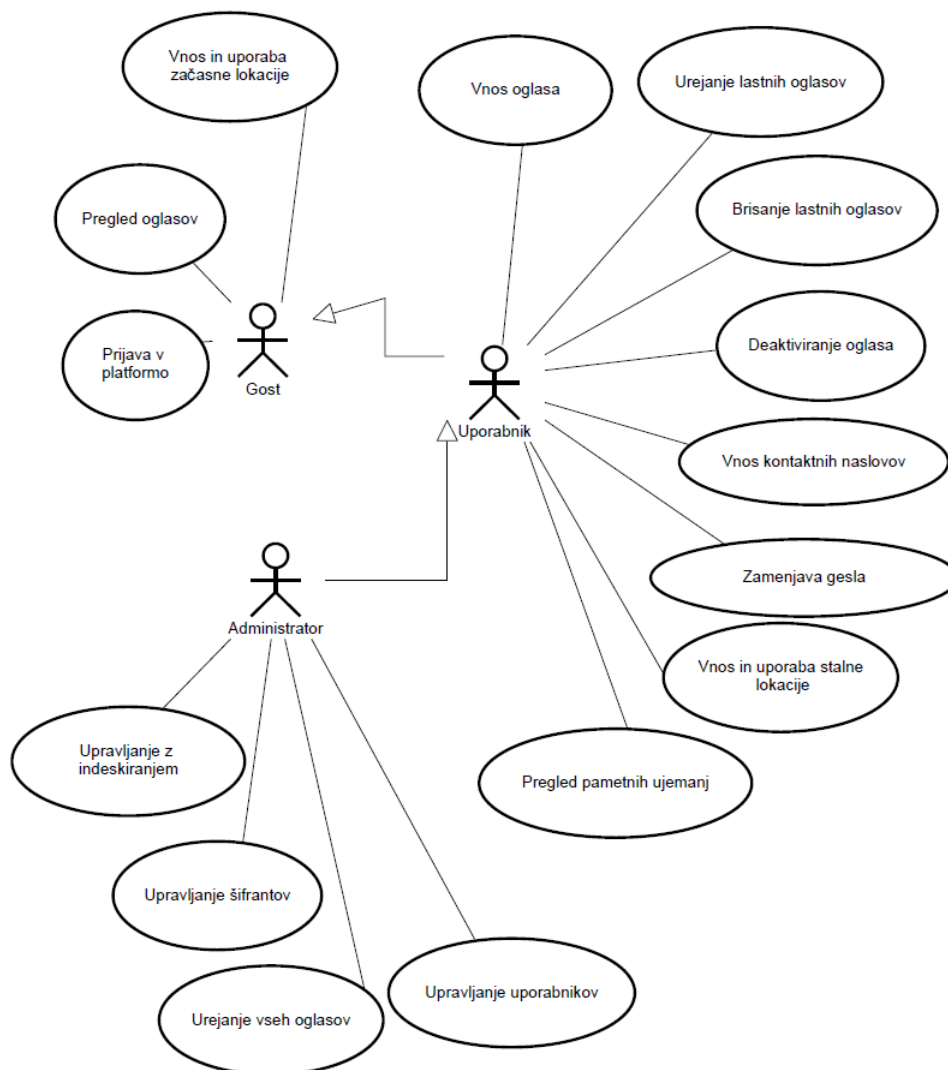
Ideja platforme je uporabnikom zagotoviti enostaven način za oddajo in pregled oglasov. Oglas je obravnavan kot predmet, ki ga je mogoče **najeti**, **oddati**, kupiti, prodati ali podariti. Slednji trije so vključeni zgolj zaradi enostavnosti vključitve. Primarno vlogo smo postavili na pomen najemanja in oddajanja.

Med delom smo sledili ciljem iterativnega razvoja, ki nam je omogočal hitro napredovanje in povezovanje segmentov aplikacije. Tako smo se najprej posvetili kritičnim točkam aplikacije, za vzpostavitev delovanja, in se nato vračali nazaj ter odpravljali napake ali izboljševali obstoječo funkcionalnost.

Problem postavitve platforme smo z analizo zahtev razdelili v več segmentov funkcionalnih zahtev in specifikacij. Kritične segmente za uspešen zagon platforme smo nato zbrali v diagram primerov uporabe.

3.1 Diagram primerov uporabe

Kot se za vsak resni razvoj spodobi, smo najprej definirali, na kakšen način se bo sistem uporabljal. Izdelali smo diagram primerov uporabe, ki temelji na treh vlogah uporabnikov - administrator, uporabnik, gost. Diagram je prikazan na spodnji sliki.



Slika 3.1: Diagram primerov uporabe

Vloge gosta so omejene na pregledovanje in iskanje oglasov, prijavo v platformo in uporabo lastne začasne lokacije.

Vloga uporabnika razširja vlogo gosta. Uporabnik ima tako na voljo tudi oddajo, brisanje in urejanje lastnih oglasov, stran z uporabniškimi nastavitvami, shranjevanje stalnega naslova, upravljanje z lastnimi kontaktnimi na-

slovi, ki se prikazujejo ob oglasih, menjavo gesla in pregledovanje pametnih ujemanj, ki mu jih je sistem našel.

Administratorska vloga poleg naštetega dopušča možnost urejanja vseh oglasov, upravljanje z indeksiranjem oglasov, različnih šifrantov in uporabnikov.

Naštet spisek akcij smatramo kot dovolj velik nabor funkcij za osnovno delovanje sistema.

3.2 Načrtovanje podatkovne baze

Podatkovno bazo smo načrtovali na podlagi diagrama primerov uporabe. Za izdelavo smo uporabili MySQL Workbench, ki nam je, po končanem podatkovnem modelu, le tega preslikal v dejansko bazo MySQL.

S končnim rezultatom preslikave podatkovnega modela smo bili zelo zadovoljni, saj je bil takoj pripravljen za uporabo. Bazo sestavlja 19 tabel, ki so med seboj povezane s tujimi ključi. Tabele po abecednem vrstnem redu:

- `b_address` - Hrani podatke o naslovih uporabnikov.
- `b_address_type` - Šifrant tipov naslovov, ki se nahajajo v tabeli `b_address`.
- `b_category` - Opredeljuje katere kategorije obstajajo v sistemu. Vrhnja kategorija je poimenovana "root" in nanjo so povezane vse podkategorije. Ker je tabela vezana sama nase s povezavo ena proti mnogo, nam to omogoča enostavno vdelavo drevesne strukture.
- `b_city` - Vnaprej pripravljen seznam mest v Sloveniji. Vsaka vrstica ima tuj ključ na občino `b_municipality`. Z izbiro mesta hitro ugotovimo občino, področje in državo. Šifrant je enostavno razširljiv na druge države.
- `b_country` - Vnaprej pripravljen seznam držav.
- `b_electronic_address` - Vsebuje kontaktne naslove uporabnikov. Ti naslovi so prikazani pri vsakem uporabnikovem oglasu in se dinamično polnijo iz te tabele glede na uporabnika, ki je oglas oddal.

- `b_item` - V tabeli so shranjeni vsi oglasi, ki se trenutno nahajajo v sistemu. Tabela vsebuje vse relevantne podatke, ki služijo za prikaz oglasa: lokacija, opis, naslov, uporabnik, ki je oglas oddal, začetek veljavnosti, konec veljavnosti, kategorijo, števil ogledov in datum vnosa oglasa.
- `b_item.type` - Šifrant nam pove kakšnega tipa je lahko oglas. En tip ima lahko več oglasov, oglas pa lahko pripada le enemu tipu. Tipi so: najem, oddaja, prodaja, nakup, podaritev.
- `b_log` - Tabela sistemskih dogodkov.
- `b_municipality` - Vnaprej pripravljen seznam slovenskih občin. Če uporabnik vnese mesto se avtomatsko pri njegovih oglasih prikaže občina.
- `b_region` - V tej tabeli je seznam vnaprej pripravljenih slovenskih pokrajin za lažje filtriranje.
- `b_settings` - V tabeli so sistemske nastavitve. Ker se platforma opira na te nastavitve lahko na enem mestu hitro spremenimo obnašanje platforme.
- `b_smart_match` - V tabeli so shranjeni rezultati algoritma za pametno ujemanje oglasov. Tabela je sestavljena iz uporabnika, ki mu je oglas namenjen in številka oglasa, ki mu ga je sistem našel.
- `b_token` - Vsebuje seznam veljavnih vrednosti, ki jih platforma dodeli uporabnikom. Uporabnik brez teh vrednosti v seji ne more klicati AJAX funkcij.
- `b_user` - Tabela vsebuje seznam vseh registriranih uporabnikov in njihovih podatkov.
- `b_user_role` - Platforma pozna tri vnaprej pripravljene vloge, ki jih hranimo v tej tabeli. Uporabnik ob registraciji dobi vlogo "uporabnik". Vlogo "administrator" lahko dodeli naknadno, ko je uporabnik prijavljen. Neprijavljeni obiskovalci pa dobijo vlogo gost.
- `b_user_role_linker` - Tabela povezuje tabelo uporabnikov in tabelo vlog.

Poglavje 4

Implementacija

4.1 Arhitektura

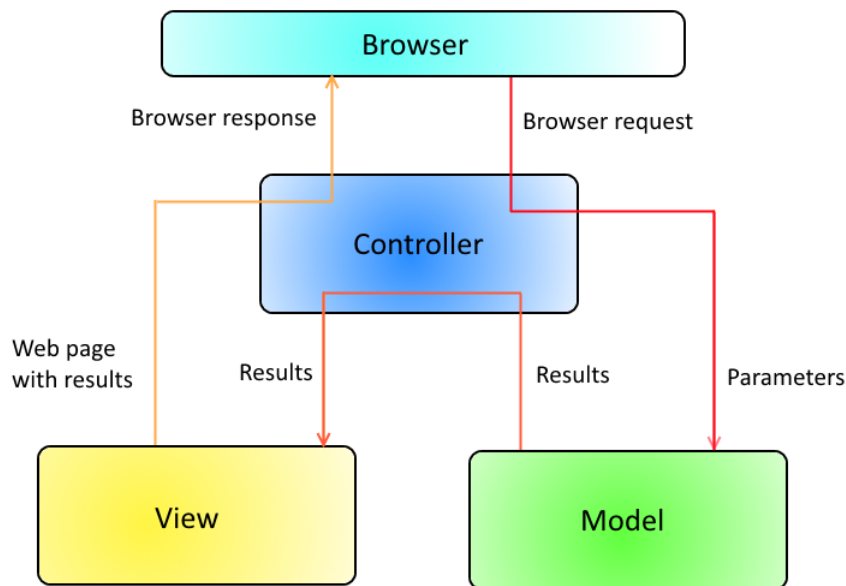
Pri implementaciji funkcionalnosti, iz diagrama primerov uporabe, smo sledili arhitekturi MVC, ki je prikazan na sliki 4.1. Glavni namen te arhitekture je preprečevanje mešanja programske kode iz različnih področij. Celota je tako smiselno razdeljena v tri elemente, ki med seboj komunicirajo preko vmesnikov:

- Kontroler (ang. Controller)

nadzira izvajanje prejetega zahtevka s strani brskalnika in poskrbi za odgovor. Lahko rečemo, da se v njem nahaja vse poslovna logika naše aplikacije. Kot povezovalni element med modelom in pogledom preveri pravilnost zahtevka in sproži ustrezne akcije v modelu. Z rezultati, ki jih prejme, dopolni pogled in vrne le tega nazaj uporabniku.

- Model (ang. Model)

vsebuje vso potrebno logiko za upravljanje s podatki. Je element, ki povezuje aplikacijo z virom podatkov, pa naj gre za podatkovno bazo, spletne storitve preko JSON-a, tekstovno datoteko ali druge. V njem so vse metode za pridobivanje in zapisovanje podatkov, zato mimo njega ne smemo iti. Na tem mestu je priporočljiva tudi kontrola pravilnosti podatkov preden se le ti, shranijo.



Slika 4.1: Model - View - Controller arhitektura

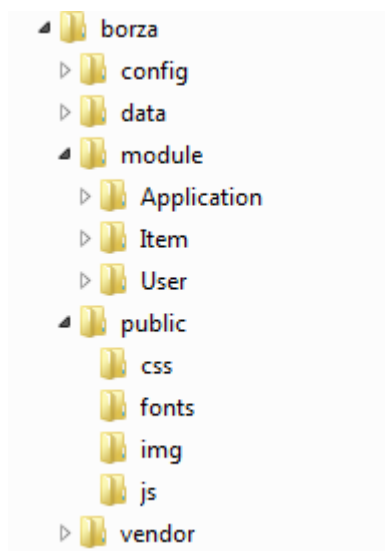
- Pogled (ang. View)

Kot že samo ime pove, se v tem modelu nahaja vsa potrebna koda za izris grafičnega vmesnika uporabniku. Tu se uporabijo vsi predhodno pridobljeni podatki, ki jih kontroler posreduje iz enega ali več modelov. Izhodna vsebina je lahko statična spletna stran, lahko pa tudi dinamično generiran podatkovni vir kot je recimo JSON ali pa XML.

4.2 Struktura datotečnega sistema

Slika 4.2 prikazuje privzeto razporeditev datotek in map v ogrodju Zend Framework 2. Že iz začetne postavitve smo prisiljeni, da razvrščamo datoteke na pravilna mesta. Sama struktura je za nepoznavalce nekoliko zapletena, vendar

se ob delu hitro naučimo, kam katera datoteka spada. Prostor za sistemske nastavitve je strogo ločen od nastavitev modulov.



Slika 4.2: Začetna struktura datotečnega sistema pri ZF2

Kratek opis pomena posamezne lokacije:

- **Config**
To je mapa v kateri so nastavitve jedra aplikacije. V njej smo določili osnovne značilnosti delovanja aplikacij in module, ki naj jih uporablja.
- **Data**
V njej so shranjeni indeksi, predpomnjene vsebine, naložene datoteke, seje, poročila o napakah in drugo.
- **Module**
Vsebuje vso uporabniško kodo, ki smo jo razvili za delovanje sistema. Koda je smiselno organizirana v posamezne sklope kot so: Upravljanje in delo z uporabniki, upravljanje oglasov, administrativni modul, modul za spletne storitve. Vsak modul ima tudi vnaprej definirano strukturo, ki določa lokacijo kontrolerjev, modelov in pogledov.

- Public

To je edina javno dostopna mapa preko brskalnika. V njej se nahaja splošno uporabljene stvari za prikaz spletnega vmesnika, kot so vrsta pisave, CSS oblike, JavaScript in jQuery knjižnice. Strežnik vso ostalo vsebino za delovanje aplikacije jemlje iz map, ki niso javno dostopne. To izboljšuje varnost in preprečuje neposredno krajo vsebine.

- Vendor

V tej mapi se nahajajo ključne datoteke ogrodja in njegove razširitve. Vsebine mape ne smemo spreminjati, saj bi bile spremembe ob naslednji nadgradnji jedra izgubljene.

4.2.1 Opis vsebine modulov

Za delovanje aplikacije smo razvili oz. dogradili pet modulov:

- Administracijski modul (ime Administration)

Vsebuje vse potrebne funkcije in poglede za upravljanje strani. V njem lahko upravljamo z uporabniki in šifranti ter izvedemo nekatera sistem-ska opravila, kot je kreiranje, optimiziranje in brisanje indeksov. Modul je dosegljiv preko navigacijskega gumba "Upravljanje strani", ki je na voljo le uporabnikom z vlogo administrator.

- Aplikacijski modul (Application)

Skrbi za prikaz navigacijske vrstice in vnos začasne lokacije uporabnika. Poleg tega je v njem definiran tudi osnovni videz aplikacije in obvezno preverjanje parametrov ob nalaganju strani (seja, nastavitve, parametri v URL naslovu). Prevzema tudi vso operacijo z AJAX klici.

- Modul za oglase (Item)

Njegovo delovanje je usmerjeno v delo z oglasi. Omogoča dodajanje, urejanje, brisanje, aktiviranje, deaktiviranje, iskanje in razvrščanje oglasov. V njem so tudi funkcije za prikaz zadnjih in največkrat ogledanih oglasov, ki ga uporablja Aplikacijski modul.

- Modul za servise in šifrante (Codetable)

Cilj modula je namenjen posredovanju splošnih vrst šifrantov, ki se

lahko uporabljajo v kateremkoli modulu (mišljeno tudi za razširitve aplikacije v prihodnosti). Nudi dostop do kategorij, vrst elektronskih in stalnih naslovov, šifranta mest, občin, pokrajin in držav ter preverjanje in dodeljevanja tokenov (Token je niz znakov, ki se dodeli uporabniku ob dostopu do platforme. Brez veljavnega tokena ni mogoče dostopati do AJAX klicev.)

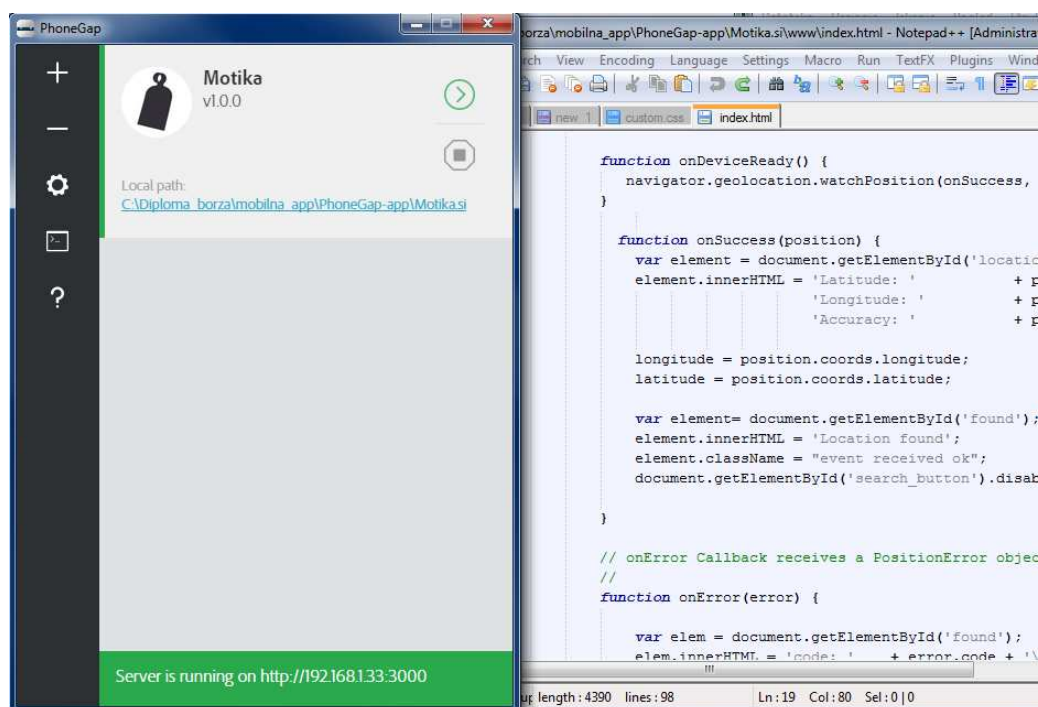
- Uporabniški modul (User)

Upravljanje podatkov uporabnika, določanje stalne lokacije uporabnika, pregled uporabnikovih oglasov, menjava gesla, menjava elektronskega naslova, prijava, registracija in odjava iz platforme so le nekatere funkcije, za katere skrbi ta modul.

4.3 Mobilna aplikacija

Za razvoj mobilne aplikacije smo uporabili HTML, CSS in JavaScript. S pomočjo aplikacije PhoneGap, katere vmesnik je prikazan na sliki 4.3 levo, smo izkoristili možnost takojšnjega predogleda vsebine na mobilni napravi. PhoneGap na računalniku deluje kot spletnih strežnik, ki sestavi izvirno kodo (Slika 4.3 desno) v paket in ga posreduje PhoneGap mobilni aplikaciji na tablici oz. mobilnem telefonu. Ta ga je skoraj v realnem času prikazal na telefonu, kar je naredilo naš razvoj bliskovit. S pomočjo navodil za uporabo razširitev smo, v zelo kratkem času, iz telefona pridobili podatke o lokaciji in sestavili grafični vmesnik. Le ta je prikazan na sliki 4.4.

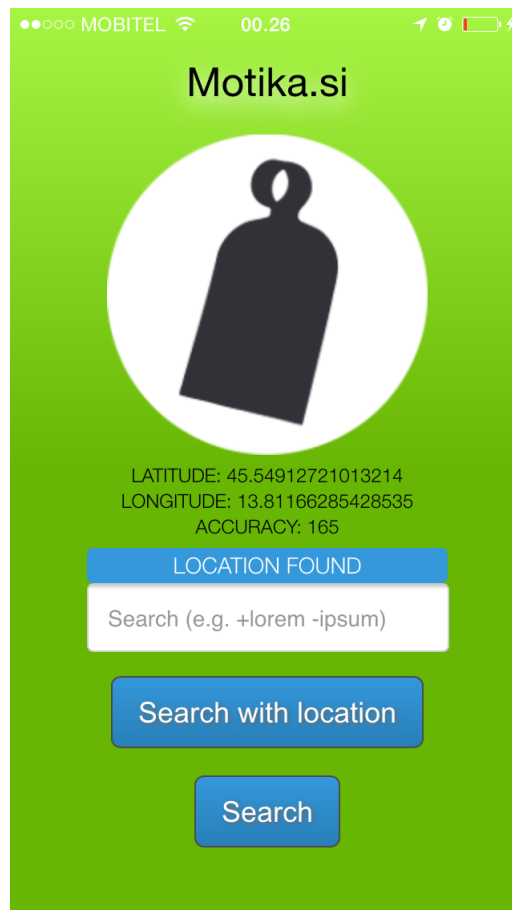
Ker je naša platforma odzivna in prilagojena na vse velikosti ekranov, smo le to izkoristili za tudi mobilno aplikacijo. Sam grafični vmesnik mobilne aplikacije je preprost. Uporabniku omogoča vnos iskanih besed in proženje iskanja s klikom na enega izmed dveh gumbov. Prvi omogoča iskanje s trenutno lokacijo uporabnika (če je le to mogoče pridobiti iz naprave, kar je razvidno iz podatkov pod logotipom), drugi pa nudi iskanje brez lokacije - zgolj z uje-manjem oglasov z iskanimi besedami. Ob kliku na izbrani gumb se znotraj aplikacije odpre vgrajeni brskalnik, ki prikazuje zadetke iz platforme. Tu je možna tudi prijava in razvrščanje ter vse druge akcije, ki jih lahko opravimo, če obiščemo platformo preko brskalnika na osebнем računalniku.



Slika 4.3: PhoneGap in del izvirne kode za pridobivanje položaja uporabnika

Dobra lastnost PhoneGap-a je, da lahko z isto izvirno kodo pokrijemo tako Android, Windows kot tudi iOS mobilni sistem. Ko smo bili z rešitvijo zadovoljni, smo jo naložili na spletno stran <https://build.phonegap.com/>. V nekaj minutah jo je sistem prevedel v pravo mobilno aplikacijo, ki bi jo lahko objavili v izbranih mobilnih trgovinah.

Za celotno izdelavo mobilne aplikacije tako nismo potrebovali razvojnih orodij proizvajalcev, izvirna koda pa je bila ista za vse tri mobilne sisteme. Izdelava mobilne aplikacije na tak način je zelo enostavna in hitra, vendar ima tudi slabe lastnosti. Primer le-te je videz menijev, gumbov in drugih navigacijskih oblik, ki niso oblikovane po zgledu celotnega mobilnega sistema na napravi. To lahko nekatere uporabnike odvrne od uporabe, saj niso vajeni takih vmesnikov.



Slika 4.4: PhoneGap mobilna aplikacije prikazuje izgled in omogoča uporabo naše izvirne kode v skoraj realnem času

Poglavje 5

Opis uporabniškega vmesnika

Izgled prve strani platforme je prikazan na sliki 5.1.

5.1 Administratorski del

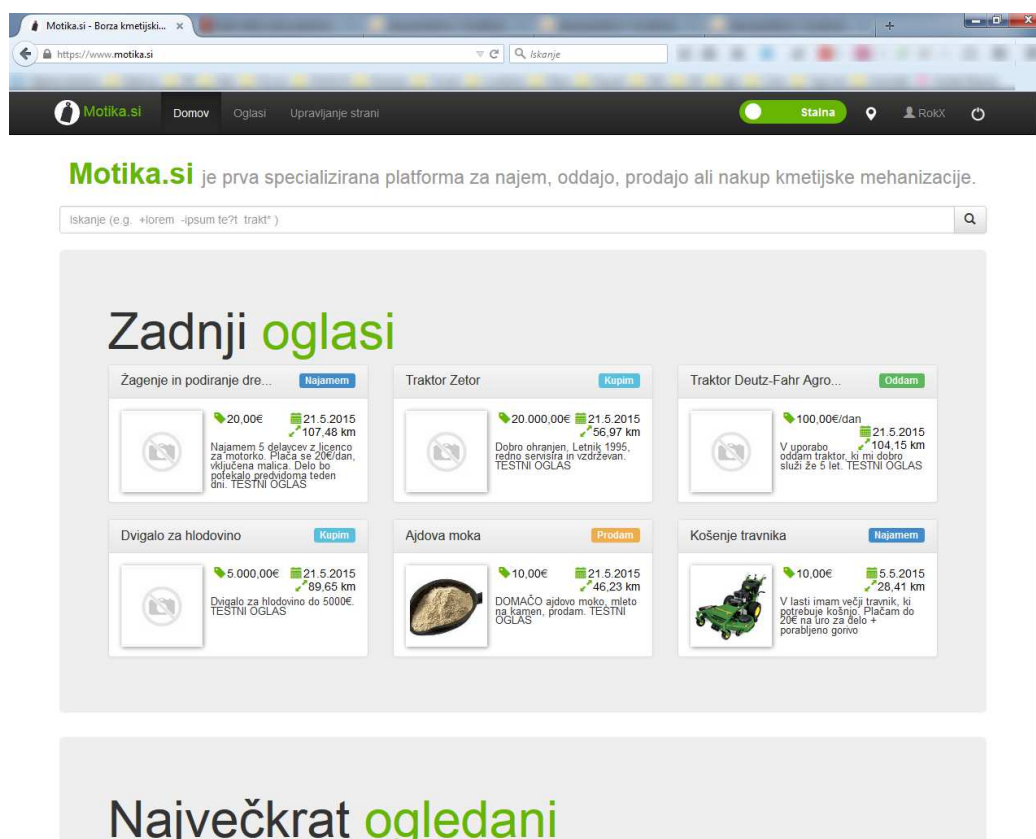
Administratorski del aplikacije je namenjen spremljanju delovanja platforme, upravljanju s šifranti in izvajanju vseh potrebnih del za nemoteno delovanje platforme. Zaslonska maska prikazana na sliki 5.2 prikazuje videz administratorskega vmesnika, ki nudi dostop do različnih akcij.

Preko globalnih nastavitev lahko spreminjamo podatke za povezavo z bazo. Gumba kategorije in uporabniki omogočata dodajanje, urejanje in brisanje kategorij in uporabnikov. Pod statistiko je pregled števila vseh oddanih oglasov, število oglasov po dnevih v zadnjem tednu, zgodovina trajanj obdelave za pametno ujemanje in podobni podatki. Gumb iskalni indeks nam omogoča ponovno izgradnjo indeksa ali njegovo optimizacijo.

5.2 Uporabniški del

Uporabniški del platforme se začne s prvo stranjo (slika 5.1) na kateri so prikazani zadnji dodani oglasi, največkrat ogledani oglasi, povezava do mobilnih aplikacij, nekaj besed o projektu in kam se lahko uporabnik obrne po pomoč.

Na vrhu zaslona je vedno prisotna navigacijska vrstica, ki uporabnika vodi



Slika 5.1: Prva stran borze za kmetijsko mehanizacijo v brskalniku Firefox

po podstraneh. V primeru prijave se navigacijska vrstica ustrezno dopolni s povezavami glede na vlogo uporabnika. Na prvi strani imamo še pregled 6 zadnjih oddanih oglasov in 6 največkrat ogledanih aktivnih oglasov.

Takoj pod navigacijsko vrstico je polje za vnos iskanih besed. Po sprožitvi iskanja smo preusmerjeni na podstran z oglasi (Slika 5.3). Prikažejo se nam oglasi, ki se ujemaajo z iskalnimi besedami. Oglase je mogoče razvrščati po razdalji (v primeru, da je uporabnik določil svojo lokacijo), naslovu, ceni in datumu oddaje oglasa. Na levi delu strani z oglasi lahko najdemo polje s kategorijami, ki nam omogoča podrobnejše filtriranje.

Oglasi v seznamu so vidno ločeni v okvirje. Poleg naslova, cene in datuma objave je v okvirju prikazana tudi pomanjšana slika oglasa ter tip oglasa.




Slika 5.2: Maska administratorskega vmesnika


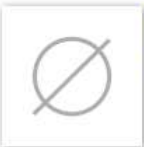


S klikom na naslov ali sliko oglasa smo preusmerjeni na podrobnosti izbranega oglasa (Slika 5.4). Poleg že prej navedenih podrobnosti se prikažejo tudi kontaktni podatki o uporabniku, ki je oglas oddal, ter lokacija oglasa na zemljevidu. V primeru, da smo določili začasno oz. stalno lokacijo pa se nam na zemljevidu izriše pot med izbrano lokacijo in oglasom. V zgornjem desnem kotu podokna z lokacijo lahko vidimo tudi prikazane podatke o cestni razdalji in predvideni čas potovanja z avtomobilom.

Zemljevid je interaktiven, zato ga lahko povečamo/pomanjšamo po svojih željah, premikamo ali preklopimo v satelitski pogled, ki nam prikaže pravo sliko površja.

[Dodaj nov oglas »](#)



Oglasi: **5** Sortiranje: [Razdalja](#) ✕ | [Naslov](#) | [Cena](#) | [Datum oddaje](#)

<div>Novi kombajn Najamem</div> <div><div>50,00 € 28.4.2015 21,78 km</div></div> <div>342343</div>	<div>Kombajn Najamem</div> <div><div>5.000,00 € 28.4.2015 44,47 km</div></div> <div>Malo rabljen, dosti viden</div>
<div>Kombajn Najamem</div> <div><div>3.000,00 €/dan 29.4.2015 48,70 km</div></div>	<div>Kombajn zetor 200 Oddam</div> <div><div>2.000,00 €/dan 1.6.2015 55,33 km</div></div>

Slika 5.3: Oglasi razvrščeni glede na razdaljo od uporabnikove lokacije

Motika.si

Domov

Oglasi

Začasna

Prijava

Podrobnosti oglasa


Vleka hlodovine

Najamem

40,00 €/uro

4.6.2015

Gozdarske klešče





Po poseku potrebujem pomoč pri vleki hlodovine iz gozda.

Plačam 40€/uro + poravnam stroške goriva.

TESTNI OGLAS

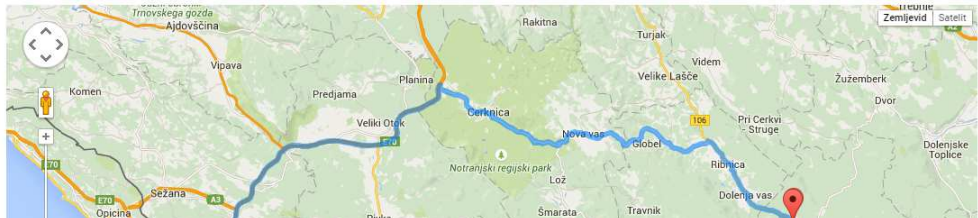
Uporabnik

Uporabniško ime: RokX

Contact:
 rokx@gmail.com
 041 714 145

Lokacija

105 km 1 h 39 min



Slika 5.4: Stran s podrobnostmi oglasa

Poglavje 6

Uporaba lokacije in pametno ujemanje oglasov

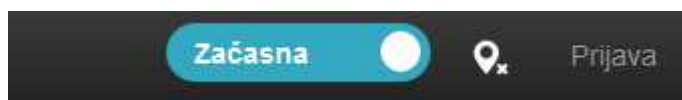
6.1 Izkoriščanje lokacije

Uporaba lokacije je dodana vrednost platforme, saj je kmetijska mehanizacija večinoma omejena z dosegom. Uporabnik s pomočjo lokacije hitreje najde primerne oglase zanj.

Podatke o lokaciji lahko platforma pridobi iz več virov. Najhitrejši je vtičnik za pozicioniranje, vgrajen v HTML5. Podpirajo ga vsi modernejši brskalniki, vendar je lahko zaznavanje lokacije v nekaterih primerih zelo natančno. Z dodatno logiko smo omogočili uporabniku spreminjanje samodejno določene lokacije. Drugi način je prikaz zemljevida, na katerega mora uporabnik klikniti za izbiro svoje lokacije. Natančnost je tu popolnoma odvisna od uporabnika. Tretji način je dosegljiv le preko mobilnih telefonov in tablic. Lokacijo mobilna aplikacije prebere iz GPS senzorjev naprave. Možnost uporabe le tega je odvisna od prostora, v katerem je uporabnik naprave in od same kakovosti senzorjev. V primeru odprtih prostorov je zaznavanje lokacije znatno hitreje in natančneje kot znotraj stavb.

Platforma omogoča sočasno uporabo dveh različnih lokacij. Kot gost lahko uporabimo le začasno lokacijo, ki jo moramo ob vsakem obisku strani znova določiti. Prijavljeni uporabnik pa lahko določi stalno lokacijo, ki bo v uporabi

takoj, ko se le ta prijavi v platformo. V primeru, da je uporabnik prijavljen, lahko uporabi tudi začasno lokacijo. Ko sta sočasno določeni tako stalna kot začasna lokacija, ima uporabnik na voljo izbiro, katero lokacijo bo uporabljal. To izbiro smo ponudili v obliki drsnika, ki je prikazan na zgornjem desnem robu vmesnika platforme, kot je prikazano na sliki 6.1.



Slika 6.1: Drsnik za izbiro med začasno in stalno lokacijo

Zraven drsnika je tudi ikona, s katero lahko izbrišemo začasno lokacijo. V primeru, da začasna lokacija ni določena, se ikona spremeni v akcijo za določanje začasne lokacije.

Pri računanju razdalj si pomagamo na dva načina. Enostavnejši način je uporaba Haversove enačbe [19], s katero izračunamo razdaljo med dvema točkama na sferi. S pomočjo podatkov o dveh točkah in polmerom sfere nam vrne razdaljo med njima, z upoštevanjem poti po sferi. Implementacija enačbe je prikazana na sliki 6.2 in jo uporabljamo pri hitrem izračunavanju razdalj med iskanjem oglasov. Drugi način je prikaz razdalje na podlagi cestnih povezav. Tu se ne moremo zanašati na enačbe, saj so ceste stvar, ki se s časom spreminja. Zaradi kompleksnosti težave smo se obrnili na Googleove zemljevide, ki se stalno posodabljaajo. Izkoristili smo njihove spletne storitve, kar nam je omogočilo prikaz cestne razdalje na zemljevidu (Slika 6.3). Storitev kličemo s podatki dveh točk, vrača pa nam poligon povezanih točk, iz katerih je sestavljena pot.

Ob kliku na oglas se nam pot prikaže na platformi s točno cestno razdaljo in predvidenim časom potovanja.

Oglasi so med iskanjem najprej filtrirani glede na pomembnost. Iz množice vseh aktivnih oglasov tako dobimo le podmnožico relevantnih oglasov, za katere izračunamo razdaljo od uporabnika. Prikazani oglasi so razvrščeni, tako da so najbližji relevantni oglasi na vrhu.


```
/**
 * Haversine formula
 *
 */
public function getAerialDistance($latitude1, $longitude1, $latitude2, $longitude2) {
    $earth_radius = 6371;

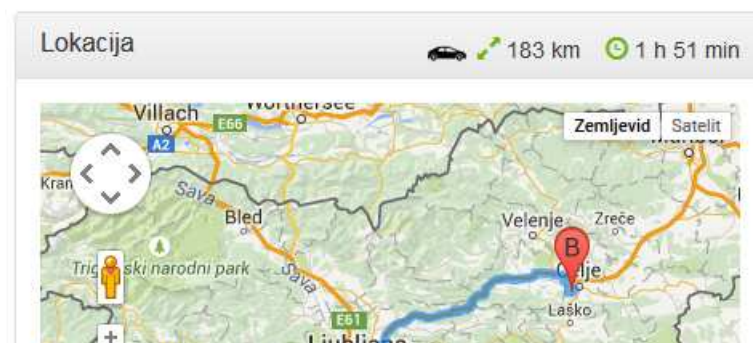
    $dLat = deg2rad($latitude2 - $latitude1);
    $dLon = deg2rad($longitude2 - $longitude1);

    $a = sin($dLat/2) * sin($dLat/2) + cos(deg2rad($latitude1)) * cos(deg2rad($latitude2))
        * sin($dLon/2) * sin($dLon/2);
    $c = 2 * asin(sqrt($a));
    $d = $earth_radius * $c;

    //echo $d;

    return number_format($d,2,',','');
}
```

Slika 6.2: Implementacija Haversove enačbe v PHP jeziku



Slika 6.3: Prikaz poti pod podrobnostmi oglasa

6.2 Pametno ujemanje oglasov

Pametno ujemanje oglasov je naziv za algoritem, ki, prijavljenemu uporabniku, na podlagi njegovih aktivnih oglasov, poizkuša najti druge primerne oglase v sistemu. Kot primerne smatramo oglase nasprotnega tipa (najem - oddaja, nakup - prodaja), ki so od uporabnikovega oglasa oddaljeni manj kot 50 km. Razdalja je trenutno fiksirana v kodi, vendar bomo za njo naredili prostor na uporabniške profilu. Pogoj za delovanje algoritma je vsaj en akti-

ven oglas tipa najem, oddaja, nakup ali prodaja.

Delovanje algoritma lahko strnemo v naslednje korake:

1. Najdemo vse uporabnike z aktivnimi oglasi.
2. Za vsakega uporabnika iz 1. točke najdemo vse njegove aktive oglase.
3. Naslov vsakega najdenega oglasa iz 2. točke podamo iskalnemu mehanizmu, ki nam vrne relevantne zadetke.
4. Pri zadetkih preverimo:
 - relevantnost more biti večja kot 0.75 (razpon od 0 do 1),
 - zadek ne sme biti isti, kot iskani oglas
 - zadek ne sme biti od istega uporabnika
 - razdalja med zadetkom in oglasom ne sme biti večja kot 50 km
 - zadek in oglas morata biti nasprotnega tipa
5. V primeru, da zadek ustreza merilom, ga shranimo v podatkovno bazo.
6. Postopek ponavljamo za vsakega uporabnika z aktivnim oglasom.

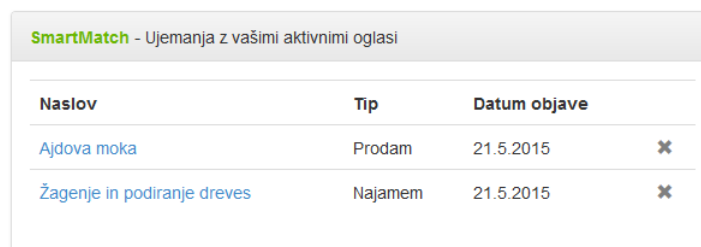
Delovanje pametnega ujemanja oglasov je postavljeno v metodo kontrolerja `JobController`. Dostop do navedenega kontrolerja je dovoljen le preko konzole, kar pomeni, da algoritma ne moremo sprožati preko spletne strani. Pametno ujemanje lahko ročno poženemo z ukazom iz konzole: **`php public/index.php job smart_match`**. Parametra `job` in `smart_match` določata akcijo, ki naj se sproži.

Dobra stran proženja akcij preko konzole je, da lahko proženje periodično ponavljamo in preprečimo, da bi se hkrati izvajali dve instanci iste akcije.

Po končani obdelavi (akciji), se podatki prikažejo na profilu uporabnika v sekciji *Vaši oglasi*, kot je vidno na sliki 6.4. Uporabnik ima tu možnost zadetke skriti, da se mu v prihodnje ne prikažejo več.

S klikom na naslov oglasa se pokažejo podrobnosti najdenega oglasa.

Vaši oglasi



SmartMatch - Ujemanja z vašimi aktivnimi oglasi			
Naslov	Tip	Datum objave	
Ajdova moka	Prodaj	21.5.2015	X
Žaganje in podiranje dreves	Najem	21.5.2015	X

Slika 6.4: Prikaz pametnih ujemanj na uporabnikovi strani

Rezultati testiranja algoritma so identični iskanju z iskalnimi pojmi le, da so pri pametnem ujemanju upoštevani nekoliko strožji pogoji. V bazo 40. oglasov smo dodali štiri oglase nasprotnih tipov. En par tipa najem/oddaja in en par tipa nakup/prodaja. Oglase smo porazdelili med štiri naključne uporabnike. Naslovi oglasov so bili nastavljeni tako, da je bila relevantnost večja kot 75 odstotkov.

Proces pametnega ujemanja je na tako majhni množici oglasov trajal le nekaj sekund in našel le dva zadetka od štirih. Po pregledu posameznega uporabniškega profila smo ugotovili, da so bili najdeni oglasi pravilno prikazani pri vsakem uporabniku. Pri paru oglasov, ki ga algoritem ni našel, se je izkazalo, da je razdalja med oglasoma nasprotnega tipa bila večja kot 50 km. Nehote napačno nastavljena lokacija oglasov in predhodno omenjena pravilna najdba sta potrdili pravilnost delovanja našega algoritma.

Poglavje 7

Postavitev strežnika in zagon aplikacije

7.1 Postavitev strežnika

Borzo smo postavili na zasebni strežnik, kjer teče Windows Server 2008 R2, Apache 2.4, PHP 5.5 in baza MySql Server 5.6. Tako baza kot sama koda aplikacije je nastavljena na dnevno izdelavo inkrementalnih varnostnih kopij. Proces za pametno ujemanje oglasov se sproža vsako uro preko *Razporejevalnika opravil* (ang. Task Scheduler).

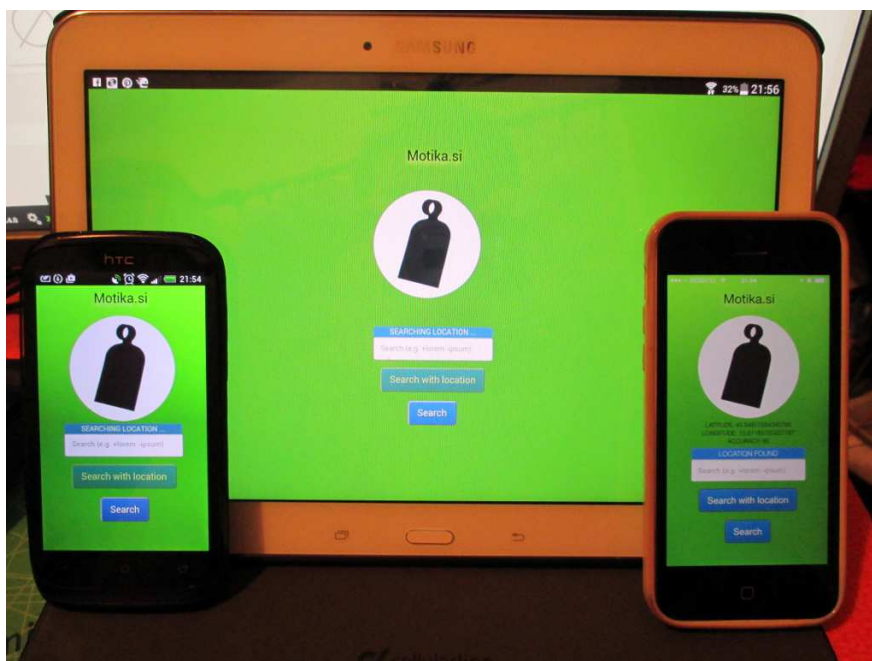
Komunikacijo med strežnikom in uporabniki smo zaradi varovanja osebnih podatkov zaščitili s pomočjo certifikata za spletne strežnike, ki je, za obdobje enega leta, brezplačno na voljo preko podjetja StartSSL (<https://www.startssl.com/>). Varno povezavo in nastavitve spletnega strežnika smo testirali preko portala Qualys SSL Labs na naslovu <https://www.ssllabs.com/ssltest/>.

7.2 Subversion in objava aplikacije

Skozi proces razvoja in implementacije smo uporabljali Subversion repozitorij kode. Omogočal nam je hiter pregled opravljenega dela in večkrat poskrbel, da nismo izgubili pomotoma izbranih datotek. Od prvega prispevka v repo-

zitorij, ki je bil narejen 26. 4. 2015, je bilo narejenih 25 korakov, kar nas je pripeljalo do delujoče različice borze.

Platformo za borzo kmetijske mehanizacije smo poimenovali **Motika.si** in jo javno objavili, 10. 5. 2015, na naslovu <https://www.motika.si>. Mobilne aplikacije zaradi stroškov razvijalskih licenc nismo objavili v trgovinah z mobilnimi aplikacijami, smo jih pa testirali na več napravah. (Slika 7.1)



Slika 7.1: Prikaz mobilne aplikacije na več napravah. (od leve proti desni) HTC Desire X, Samsung Galaxy Tab 4, iPhone 5C

Poglavje 8

Sklepne ugotovitve

V diplomskem delu smo prikazali celoten potek razvoja platforme za borzo kmetijske mehanizacije. Začeli smo s predstavitvijo uporabljenih orodij in tehnologij, ki smo jih pri svojem delu uporabili. Nadaljevali smo z analizo zahtev in diagramom primerov uporabe pa vse do implementacije platforme v ogrodju Zend Framework 2. Videli smo, kako je ogrodje zasnovano in ga razširili z lastnimi moduli. Razvili smo še mobilno aplikacijo in podali kratka navodila za uporabo platforme. Za zaključek smo se obrnili še na dodani vrednosti platforme (upoštevanje lokacije in pametno ujemanje oglasov) in potek objavljanja aplikacije na strežnik za javno uporabo.

Koncept MVC arhitekture je težak za začetnike, vendar nas je sama struktura map v uporabljenem ogrodju vodila po pravi poti. Ker smo svojo idejo dobro pretvorili v funkcionalne zahteve, je potek razvoja, po začetnem spoznavanju ogrodja, minil hitro in učinkovito. Vmesnik platforme smo sproti testirali v različnih brskalnikih kot tudi na mobilnih napravah. Zaradi optimizacije le tega, na vse velikosti zaslonov, in uporabo dobrega razvijalskega orodja za mobilne naprave smo za mobilno aplikacijo porabili manj časa od predvidenega.

V primerjavi z obstoječimi storitvami, kot so spletni portali <http://www.bolha.com/>, <http://kmetijskioglasnik.si/> in <http://www.stroji.net/>, ponuja naša platforma nove razsežnosti oglaševanja. Uporaba lokacijskih storitev je v drugih portalih nezaznavna, kar daje naši platformi izjemno prednost. Priporočilni sistem smo opazili le na bolha.com, le ta pa se skli-

cuje na naša predhodna iskanja. Naša platforma opravlja v ozadju pametno iskanje, ki ga drugi portali trenutno ne ponujajo. Prednost naše platforme je tudi osredotočenost na kmetijsko mehanizacijo, saj bi lahko filtriranje v prihodnosti zelo podrobno razvejili. Večji poudarek smo postavili na oglase tipa najem/oddaja, kar pospešuje sodelovanje kmetov in prodor spletnih tehnologij v to področje. Svoje stroje lahko tako bolje izkoristijo z oddajo v najem ali ponudijo celotno storitev. S pametnim ujemanjem oglasov in uporaba lokacijskih podatkov smo naredili konkurenčno prednost in razbremenili delo uporabnika. Odslej ni več potrebno iskanje med neskončnimi oglasi, ampak to delo za vas opravi sistem.

8.1 Možnosti nadaljnjega razvoja

Mnogo idej za nove funkcionalnosti se nam je pojavilo med samo implementacijo. Kot najbolj zanimive predstavljamo naslednje:

- mobilno aplikacijo bi lahko razširili z oddajanjem oglasov tako, da bi pri tem izkoristili GPS senzorje ter fotoaparata na mobilni napravi;
- platforma bi lahko uporabnike takoj obveščala o novih zadetkih pametnih ujemanj oglasov preko SMS sporočil, elektronske pošte in/ali potisnih sporočil (ang. *push notifications*);
- spletni del platforme bi lahko omogočal osnovno urejanje (obrezovanje, megljenje, izenačevanje barv) naloženih slik in tudi možnost nalaganja več slik za posamezni oglas.

Cilje diplomske naloge smo skladno z zahtevami in pričakovanju omejili in uspešno uresničili. Možnosti nadaljnjega razvoja platforme pa so praktično neomejene.

Slike

3.1	Diagram primerov uporabe	10
3.2	Podatkovni model v programu MySQL Workbench	12
4.1	Model - View - Controller arhitektura	16
4.2	Začetna struktura datotečnega sistema pri ZF2	17
4.3	PhoneGap in del izvirne kode za pridobivanje položaja uporabnika	20
4.4	PhoneGap mobilna aplikacije prikazuje izgled in omogoča uporabo naše izvirne kode v skoraj realnem času	21
5.1	Prva stran borze za kmetijsko mehanizacijo v brskalniku Firefox	24
5.2	Maska administratorskega vmesnika	25
5.3	Oglasi razvrščeni glede na razdaljo od uporabnikove lokacije . .	26
5.4	Stran s podrobnostmi oglasa	27
6.1	Drsnik za izbiro med začasno in stalno lokacijo	30
6.2	Implementacija Haversove enačbe v PHP jeziku	31
6.3	Prikaz poti pod podrobnostmi oglasa	31
6.4	Prikaz pametnih ujemanj na uporabnikovi strani	33
7.1	Prikaz mobilne aplikacije na več napravah. (od leve proti desni) HTC Desire X, Samsung Galaxy Tab 4, iPhone 5C	36

Literatura

- [1] (2015) Notepad++, Dostopno na:
<https://notepad-plus-plus.org/>
- [2] (2015) PhpStorm IDE, Dostopno na:
<https://www.jetbrains.com/phpstorm/>
- [3] (2015) MySQL Workbench, Dostopno na:
<http://www.mysql.com/products/workbench/>
- [4] (2015) Zend Framework 2, Dostopno na:
<http://framework.zend.com/>
- [5] (2015) jQuery, Dostopno na:
<https://jquery.com/>
- [6] (2015) jQuery UI, Dostopno na:
<http://jqueryui.com/>
- [7] (2015) Bootstrap, Dostopno na:
<http://getbootstrap.com/>
- [8] (2015) PhoneGap, Dostopno na:
<http://phonegap.com/>
- [9] (2015) Apache Cordova, Dostopno na:
<https://cordova.apache.org/>
- [10] (2015) Adobe PhoneGap Build, Dostopno na:
<https://build.phonegap.com/>

-
- [11] (2015) Subversion, Dostopno na:
<https://subversion.apache.org/>
 - [12] (2015) HTML, Dostopno na:
<http://en.wikipedia.org/wiki/HTML>
 - [13] (2015) CSS, Dostopno na:
<http://www.w3.org/Style/CSS/>
 - [14] (2015) CSS, Dostopno na:
<http://sl.wikipedia.org/wiki/CSS>
 - [15] (2015) PHP History, Dostopno na:
<http://php.net/manual/en/history.php.php>
 - [16] (2015) PHP , Dostopno na:
<http://sl.wikipedia.org/wiki/PHP>
 - [17] (2015) Firebug , Dostopno na:
<http://getfirebug.com/>
 - [18] H. Paco; W. Ben, *Web Security Testing Cookbook*, Sebastopol, CA:
O'Reilly Media, Inc., 2008, ISBN 978-0-596-51483-9
 - [19] (2015) Haversine enačba, Dostopno na:
http://en.wikipedia.org/wiki/Haversine_formula